Reconhecimento de Objectos Simples

Uso de Redes Neuronais

João Nuno Boavista Taborda
Tiago André Dias Silva

27 de Maio de 2005

Trabalho realizado no âmbito da disciplina de Inteligência Artificial
do 2º Semestre, do 3º Ano, da Licenciatura em Eng. Informática e Computação
da Faculdade de Engenharia da Universidade do Porto, leccionada por
Eugénio da Costa Oliveira, Ana Paula Rocha e Luís Paulo Reis.

Faculdade de Engenharia da Universidade do Porto
Licenciatura em Engenharia Informática e Computação
Rua Roberto Frias s/n, 4200-465 Porto

Resumo

Este relatório foi elaborado para a disciplina de Inteligência Artificial, no segundo semestre do terceiro ano, do curso de Engenharia Informática e de Computação.

Documenta o o trabalho proposto para a disciplina, cujo tema, visa o uso do paradigma da inteligência artificial.

O tema abordado é a identificação de objectos simples, usando redes neuronais, são focados os objectivos e metodos, descritos os algoritmos utilizados e dificuldades e escolhas encontradas durante a realização do trabalho.

Conteúdo

1 Introdução

1.1 Objectivo

O objectivo deste trabalho consistiu na elaboração de um programa que permita, com a maior precisão possível, identificar correctamente objectos simples para os quais a sua rede neuronal tenha sido previamente treinada.

Pretendeu-se que a aplicação pudesse ser utilizada por qualquer pessoa com o mínimo conhecimento sobre o assunto, razão pela qual foi focado parte do esforço na elaboração de uma interface intuitiva.

1.2 Motivação

Esta área da inteligência artificial é das que apresenta maior potencial de atingir "a meta da área", a de replicação das funções cognitivas do ser humano, e quem sabe, ir mais além. Tudo aponta para que não exista um modelo de grande abstracção (conscientemente compreensível) que permita replicar tarefas cognitivas de qualquer complexidade conservando a capacidade de adaptação a novos problemas. A solução aparenta estar no domínio dos sistemas caóticos, na descoberta de processos de geração e selecção de sistemas que resistam a mudanças de ambiente e contexto, emulando assim o processo de selecção natural.

Embora a área de redes neuronais seja rodeada de modelos demasiados desconexos e teorias e supostas provas por vezes contraditórias, eventualmente se tornará mais unificada, tornando-se acessível a cada vez mais investigadores, e consequentemente dando muitos mais resultados satisfatórios ao longo do tempo. Com tempo desempenhará um grande papel em tornar os filmes de ficção de hoje nas banalidades de amanhã.

1.3 Descrição do projecto

O projecto toma a forma de duas aplicações, uma de cliente, que funciona como uma interface gráfica, e uma de servidor, que trata de processar as imagens enviadas pelo cliente e de responder com o seu pré-processamento de imagem e consequente classificação do objecto e do treino da rede.

2 Interface com câmara digital

A interface gráfica funciona como uma aplicação de cliente, permitindo ao utilizador ligar uma máquina fotográfica digital ao computador, e tirando com ela as fotos que quer ver classificadas. A interface trata de enviar a imagem para a aplicação de processamento e classificação e devolver a sua resposta. A elaboração da interface surgiu sobretudo pela motivação de preencher um certo vazio que existe nas aplicações desenvolvidas sobre o tema. Sendo o tema sobretudo foco de estudo e investigação académica, tende a focar-se sobretudo na elaboração de aplicações que demonstrem conceitos teóricos a pessoas que estejam dentro do assunto, tornando inacessível esta tecnologia ás pessoas que desejam ver o potencial da mesma, imediatamente e sem grande esforço, quer para testes experimentais, quer pela curiosidade. O desenvolvimento de mais aplicações destetipo, mais demonstrativas tornaria e diminuiria muito tempo perdido com investigação redundante e testes de tentativa e erro.

3 Pré-Processamento de imagem

A fim de introduzir informação de uma imagem na rede neuronal, esta é pré-processada e retiram-se algumas caracteristicas, varias combinações de filtros, transformações à imagem e informações extraidas foram pensadas e testadas.

Devido à enorme versatilidade de um objecto e sua posição, rotação, distancia e envolvente têm de ser consideradas caracteristicas bastante flexiveis, a escolha passou por tentar vectorizar caracteristicas das imagens.

3.1 Caracteristicas introduzidas na rede

As caracteristicas utilizadas são a descrição do contorno do objecto por vectores. Esta pareceu uma abordagem sensata mas, o problema surge, quando se pensa que a rede neuronal tem um numero fixo e finito de neuronios de input, mas, dependendo do objecto, o numero de vectores descritores do contorno pode variar.

De forma a garantir um numero fixo de caracteristicas introduzidas na rede o contorno de cada objecto é dividido num numero fixo de vectores de tamanho unitario (todos os vectores tentam ter o mesmo comprimento) igual ao numero de neurónios de entrada da rede neuronal.

As caracteristicas introduzidas na rede são os angulos que os vectores descrevem entre si, consegue-se assim informação independente da rotação e distancia pois os vectores relacionam-se entre si de forma igual, independentemente da distancia e rotação em relação ao ponto de observação.

3.2 Retirar Caracteristicas de uma imagem

Segue-se agora uma descrição dos filtros, transformações e algoritmos que são usados de forma a retirar, de uma foto de um objecto, as caracteristicas, a introduzir na rede neuronal.

3.2.1 Auto-Treshold

Em primeiro lugar aplica-se um treshold à foto do objeto de modo a eliminar alguns residuos indesejaveis.

O filtro treshold resume-se a, dado uma cor central, colocar a branco todas as cores com inferiores e a preto todas as superiores.

O algoritmo de encontrar o ponto ideal a aplicar o treshold é bastante simples, resume-se a, depois de calculado o histograma da imagem (predominancia de cada cor na imagem), encontrar o ponto ideal para transformar a imagem a 2 cores de modo a salientar as cores mais predominantes e esconder as outras.

Como se pode ver na imagem seguinte, uma caixa de cd's colocada em cima duma mesa, é nitido a eleminação de ruido (a cor da mesa a separação entre mesas, alguma sombra, algum reflexo).

Figura 1: Exemplo de Auto-Treshold

3.2.2 Procura da posição do objecto

Um objecto pode-se encontar em várias posições da imagem, esta parte do pre-processamento trata de encontrar o objecto mais proximo do centro, seguindo, a partir do centro da imagem, em espiral até encontrar um objecto.

Por exemplo na imagem a baixo o telemovel, apesar dos objectos à volta, é o que se encontra mais perto do centro e é isolado.
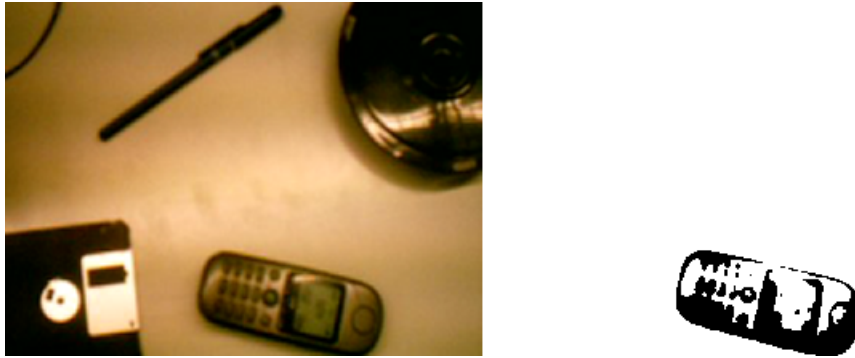
Figura 2: Exemplo de procura do objecto

3.2.3 Convex Hull

De forma a descrever o contorno de um objecto foi implementado um algoritmo bastante comum em computação gráfica e reconhecimento de padrões, o *convex hull*.

Por definição um *convex hull* é o menor poligono que, dado um conjunto disperso de pontos $\boldsymbol{S}$, consegue abranger no seu interior todos os pontos do conjunto $\boldsymbol{S}$ com o menor numero de arestas.

Eliminação de pontos Devido à complexidade, do algoritmo que descobre o *convex hull*, ser elevada antes de o encontrar eliminam-se todos os pontos que à partida são redundantes, ou seja aqueles pontos que estão rodeados por bastantes pontos.

Note-se na imagem que os pontos essenciais do contorno da disquete são mantidos.
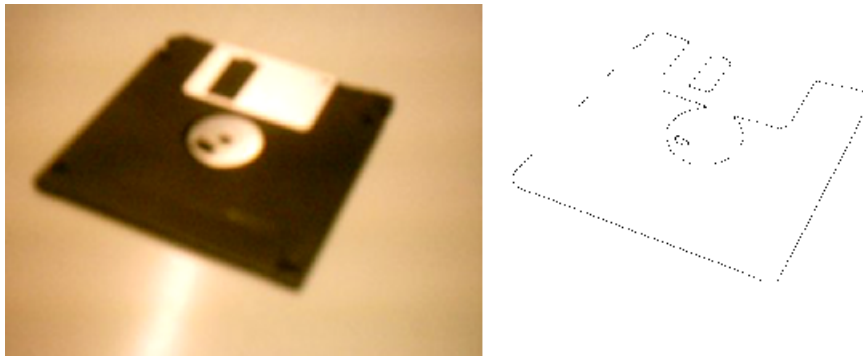
Figura 3: Exemplo de eliminação de pontos

Convex Hull O algoritmo utilizado para encontrar o *convex hull* é conhecido por *Graham Scan*, é provavelmente um dos algoritmos com menor compelxidade para encontrar um *convex hull*, a sua complexidade, apenas dependente do algoritmo de ordenação pois para pontos ordenados comporta-se linearmente, é O(n log(n)).
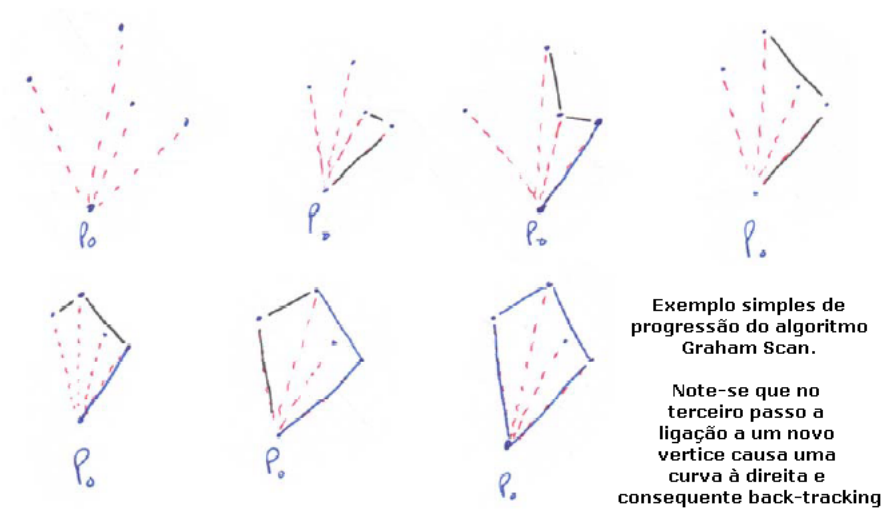
Figura 4: Explicação simples do algoritmo Graham scan

Para encontrar o *convex hull*:

- encontra-se o ponto com a menor coordenada y, se existir um empate escolhe-se a que tem menos coordenada x, toma-se este ponto como *pivot*;
- Para todos os pontos calcula-se o angulo que fazem com o *pivot*;
- Ordena-se todos os pontos pelo angulo calculado;
- Partindo do pivot liga-se ao primeiro ponto, este ao segundo...etc..;
- De cada vez que se liga um ponto a outro analisa-se o tipo de curva que faz e, quando ocorre uma curva à direita, elimina-se o ponto actual do poligono e faz-se *back-tracking;*
- Repete-se ate não sobrarem pontos desligados.

Figura 5: Exemplo de Convex hull

3.2.4 Divisão em Vectores Unitários

Seguido do *convex hull* discretiza-se o contorno em vectores de tamanho igual, seguindo o perimetro do objecto e dividindo-o.
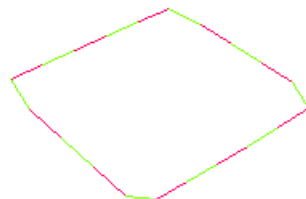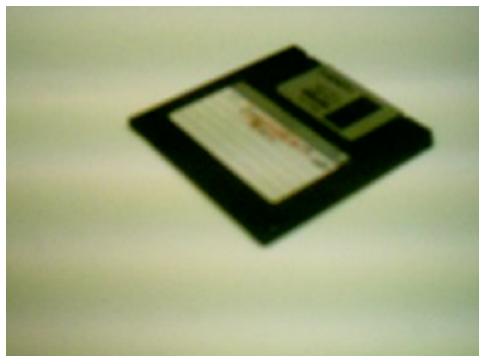
Figura 6: Exemplo de divisão em 20 vectores unitários

3.2.5 Retirar caracteristicas

Depois de encontrados os vectores que descrevem o contorno são analisados e guardados, para posterior introdução na rede neuronal, os angulos entre cada dois vectores seguidos. Criando, por exemplo para a imagem acima, esta linha de caracteristicas de angulos (note-se que devido às caracteristicas do *convex hull* nunca ultrapassam os 180º e são reduzidos a valores entre 0 e 1):

```
0.709660 0.992255 1.000000 1.000000 1.000000
0.455289 0.912571 1.000000 1.000000 0.800121
0.799569 1.000000 1.000000 0.987739 1.000000
0.501501 0.857117 1.000000 1.000000 0.974905
```

4 Rede neuronal de reconhecimento de objectos simples

Este módulo constitui o núcleo do programa, foi programado de raíz, estruturado segundo o paradigma de programação orientada a objectos e feito a pensar na sua flexibilidade. Esta arquitectura é totalmente parametrizável, o que permitiu a realização de vários testes de desempenho da rede, com diferentes arquitecturas e número de parâmetros amostrais com relativa facilidade.

Inicialmente foi tomada a decisão de utilizar três camadas escondidas, tendo em conta que este número de camadas consegue aproximar qualquer função (Hecht-Neilson 1990), no entanto, comprovou-se experimentalmente que este número poderia ser reduzido para duas sem qualquer perda de capacidade de generalização, e, tendo em conta que quantas menos neurónios menos tempo de processamento pelo algoritmo, esta escolha tornou-se permanente.

O número de neurónios óptimo a escolher para cada camada revelou-se incalculavel, de forma a chegar a uma conclusão semi-determinista, elaborou-se uma pequena aplicação para testar todas as combinações de neurónios por camada, testando para cada combinação de rede a capacidade de generalização da mesma. Depois de algumas horas de "brute-forcing" foram encontradas soluções bastante promissoras, e são utilizada na nossa implementação (ver testes experimentais).

4.1 Early Stopping

Surgiram algumas dúvidas acerca de como ensinar a rede a generalizar, alguma investigação levou à descoberta do Early Stopping Method of Training, método que se revelou muito eficaz. O método consiste na divisão dos casos de treino em duas partes, uma de treino, e uma de validação. O treino decorre normalmente com os primeiros casos, e de cinco em cinco épocas de treino, são testados os casos de validação. Na altura em que o erro de validação suba, o treino é imediatamente parado. Embora este método por si só devolva resultados satisfatórios, descubrimos experimentalmente que existem vários minimos locais, e o minimo absoluto raramente é o primeiro. Por esta razão

alteramos o método para fazer 2000 épocas de treino, e guardar o estado da rede sempre que o erro baixar. No fim, a rede neuronal com menor erro de validação é utilizada, esta variação apresentou resultados bastante melhores.
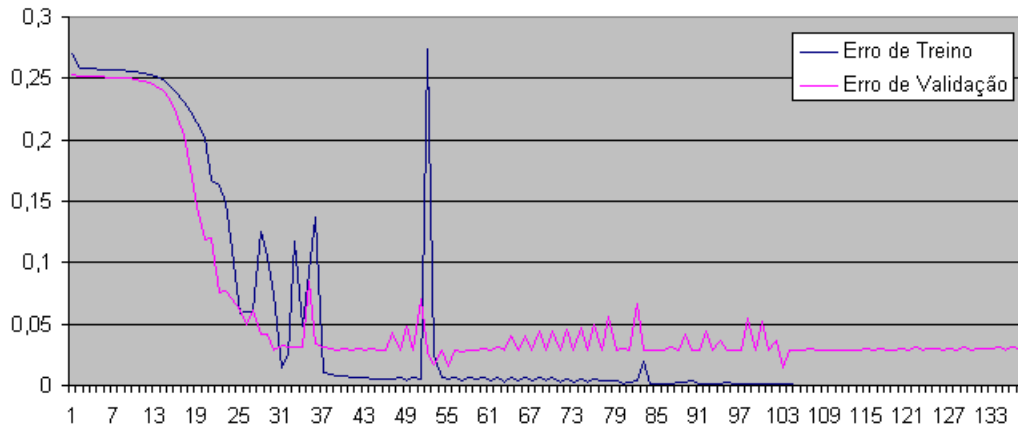
Figura 7: Gráfico de Convergencia

5 Esquemas de Representação de Conhecimento que usou

Como ilustrado anteriormente na descrição do módulo de pré-processamento de imagem e criação de arquivos de treino, foi escolhido o menor esquema de representação possível, que descrevesse as características mais relevantes de um objecto. Pensa-se que uma das característica mais relevantes de um objecto é o seu contorno, e esta é a única informação que é fornecida à rede neuronal. As características da aresta são representadas em forma dos ângulos internos dos vectores que a representam. Esta representação de conhecimento tem a grande vantagem de devolver características muito semelhantes para um dado objecto de todos os ângulos e distâncias, contribuindo então para uma grande capacidade de generalização por parte da rede neuronal.

6 Ambiente de desenvolvimento

Toda a aplicação de servidor (rede neuronal e pré-processamento de imagem) foi implementada em Visual C++ 6.0, a aplicação de cliente (interface gráfica) foi implementada C#, necessitando então da .NET Framework para ser executada. O módulo de captura de imagem da webcam, foi desenvolvido por Philip Pierce e encontrado na *internet*.

7 Avaliação do programa

O melhor programa de reconhecimento de objectos seria, obviamente, o que apresenta-se uma melhor relação de memorização/generalização. Identificando com bastante certeza os objectos para os quais fosse treinado em situações e locais diversos.

O programa ideal, um pouco utópico, seria o que conseguisse associar generalizações a derivações dessa generalização. Este programa teria que ter a capacidade de dizer que uma bola azul era uma bola azul, que uma bola vermelha era uma bola vermelha, mas que ambas eram uma bola.

Como segunda prioridade, o programa seria tanto melhor quanto maior precisão apresentasse na classificação de dois objectos bastante semelhantes.

8 Resultados Experimentais

8.1 Procura da melhor rede

Depois de alguns testes (um pouco à força) seguidos de verificações menos rigorosas encontrou-se uma boa rede com um erro de generalização bastante reduzido (resultados do teste em anexo):

- Neurónios de entrada: 32
- Neurónios da primeira camada: 25
- Neurónios da segunda camada: 3
- Neurónios de saída: 4

Pode-se então deduzir, por metodo exprimental, que para as caracteristicas introduzidas uma boa rede a primeira camada teria um número de neurónios pouco distanciado do número dos de input, e a segunda camada um número de muito perto do número de saída. Posteriores testes (com menor rigor) parecem ter comprovada esta teoria.
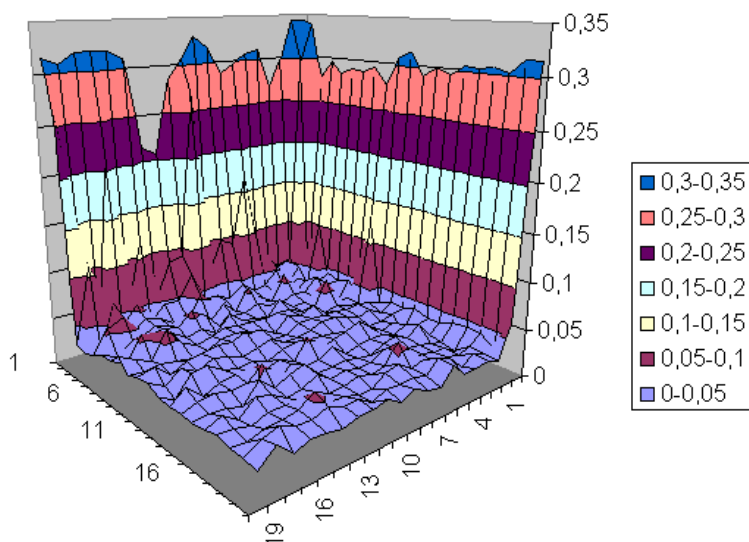
Figura 8: Grafico do erro de generalização dependendo do numero de neuronios na 1ª e 2ª camada

9 Conclusão

Embora as redes neuronais não sejam actualmente a solução para todos os problemas da investigação artificial, tudo leva a crer que eventualmente serão, as limitações que apresentam são apenas as mesmas que apresenta qualquer sistema caótico. Tal e qual uma pessoa não consegue prever as flutuações do mercado da bolsa com suficiente precisão para ficar milionária de um dia para o outro com completa consciência de todos os passos que completou para efectuar essa transição, também os resultados de uma rede neuronal são imprevisiveis sem ver o seu resultado na altura, o número de variáveis envolvidas é simplesmente colossal. No dia em que as capacidades de processamento se tornarem um problema do passado, estarão abertas as portas para possíbilitar efectuar qualquer tarefa cognitiva com redes neuronais.

Infelizmente esta área é coberta de mistério, aparentamente desenvolvida apenas por um conjunto selecto de pessoas, regra geral académicas e consequentemente com uma particular estratégia e modalidade preferidas de pensamento, que se revela primáriamente na grande preferência de

ilustração de conceitos em linguagem matemática. Este facto apresenta algum desafio na "popularização" desta tecnologia, tornando-se mais complicada do que realmente é.

O resultado do projecto foi sem dúvida alguma para além das nossas expectativas iniciais. Após um período de ingenuidade em que pensavamos que a rede neuronal conseguiria generalizar qualquer coisa, e um posterior periodo de desilusão e descrença acerca do potencial da aplicação que iríamos desenvolver, com alguma criatividade e engenho, contornamos todos os problemas que se apresentaram e desenvolvemos um sistema de reconhecimento que se comporta bastante bem, tendo em conta os parâmetros de aprendizagem escolhidos.

10 Melhoramentos

O projecto poderia ser alvo de muitos melhoramentos, não fosse a falta de informação compreensível acerca de certos temas do nosso interesse. Um dos grandes melhoramentos seria o da aplicação um algoritmo que discretizasse perfeitamente os contornos dos objectos. Encontramos duas alternativas, o "Concave Hull", e o "Gradient Vector Flow Snake Contour Detection", ambos com pouquíssima informação disponível sobre o assunto, e nenhuma dela acessível.

O algoritmo de aprendizagem poderia ser bastante acelerado com um dos muitos algoritmos que existem de optimização do "Backpropagation". Foi tentada a implementação do "RPROP", algoritmo dito como sendo até 50 vezes mais rápido que o "Backpropagation", no entanto, misteriosamente, não funcionou, embora aparentemente siga todos os passos apresentados pelos autores do algoritmo no seu artigo.

A introdução de informação sobre a distribuição de cor no objecto, na rede neuronal, talvez permiti-se fazer melhores distinções em termos, embora tenhamos implementado e testado a rede neuronal com esta informação, e até com resultados bastante satisfatórios, achamos que entraria em **conflicto** com a nossa prioridade de classificar os objectos pelo contorno, sendo a nossa prioridade a capacidade de generalização que tanto torna a rede neuronal distinta.

11 Bibliografia

11.1 Processamento de imagem

1. An Introduction to Machine Vision, James Matthews, 2000
   http://www.generation5.org/content/2000/vision.asp

2. A Basic Introduction to Image Processing, 2001
   http://www.generation5.org/content/2001/im00.asp

3. Histogram Equalization, James Matthews, 2004
   http://www.generation5.org/content/2004/histogramEqualization.asp

4. Thresholding and Segmentation, James Matthews, 2003
   http://www.generation5.org/content/2003/segmentation.asp

5. Canny Edge Detection Tutorial, Bill Green, 2002
   http://www.pages.drexel.edu/~weg22/can_tut.html

6. Open Source Computer Vision Library
   http://www.intel.com/research/mrl/research/opencv/

7. RGB/YUV Pixel Conversion
   www.fourcc.org/fccyvrgb.php

8. Computer Vision - Thinning Algorithm, R. Mukundan
   http://www.cosc.canterbury.ac.nz/people/mukundan/covn/Thin.html

9. Leaves Recognition - A leaf image recognition based on a neuronal network, LightSpeed Communications GbR
http://damato.light-speed.de/lrecog/

10. Computer Vision and Image Processing, Scott E Umbaugh, 1999

11. Graham's Scan Applet
http://www.cs.princeton.edu/~ah/alg_anim/version1/GrahamScan.html

12. Convex Hull Algorithms, Eric Eilberg, Denison University
http://personal.denison.edu/~havill/272/papers/convexhulls.pdf

13. Active Contours, Deformable Models, and Gradient Vector Flow, Chenyang Xu, J.L. Prince
http://iacl.ece.jhu.edu/projects/gvf/

14. Gradient Vector Flow: A New External Force for Snakes,Chenyang Xu, J.L. Prince
http://ieeexplore.ieee.org/iel4/83/14393/00661186.pdf

11.2 Redes Neuronais

1. Artificial Intelligence: A modern approach, Stuart Russel e Peter Norvig

2. An Introduction to Neural Networks, James Matthews,2000
http://www.generation5.org/content/2000/nnintro.asp

3. Neural Networks: Motivation, Theory and DANN, Daniel Eaton, 2000
http://www.generation5.org/content/2000/nn00.asp

4. Notes on Neural Network Learning and Training, Wan Hussain Wan Ishak, 2004
http://www.generation5.org/content/2000/NNTrLr.asp

5. Back-propagation for the Uninitiated, James Matthews, 2002
http://www.generation5.org/content/1999/bp.asp

6. Perceptrons, James Matthews, 1999
http://www.generation5.org/content/1999/perceptron.asp

7. Multi-Backpropagation Network: Concept and Modeling, Wan Hussain Wan Ishak, 2004
http://www.generation5.org/content/2004/MultiBP.asp

8. Multilayer Feedforward Network and the Backpropagation Algorithm, Samuel Hsiung, 1999
http://www.generation5.org/content/1999/nn_bp.asp

9. BP Example: XOR Net, James Matthews, 2001
http://www.generation5.org/content/2001/xornet.asp

10. Summing with Neural Networks, Stephen Tashev, 2003
http://www.generation5.org/content/2003/nn_sum.asp

11. Simple BMP File Analysis Using MLPs, Povilas Daniuðis, 2004
http://www.generation5.org/content/2004/petras.asp

12. The Face Recognition Home page, Peter Kruizinga
http://www.cs.rug.nl/~peterkr/FACE/face.html

13. Neural Networks for Face Recognition
http://www-2.cs.cmu.edu/~tom/faces.html

14. Organization of face and object recognition in modular neural network models, M.N. Dailey, G.W. Cottrell, 1998

15. Direct adaptive method for faster backpropagation learning: The RPROP algorithm, M Riedmiller, H Braun
http://leeds-faculty.colorado.edu/laguna/syst7330/rprop.pdf

12 Manual de Utilização
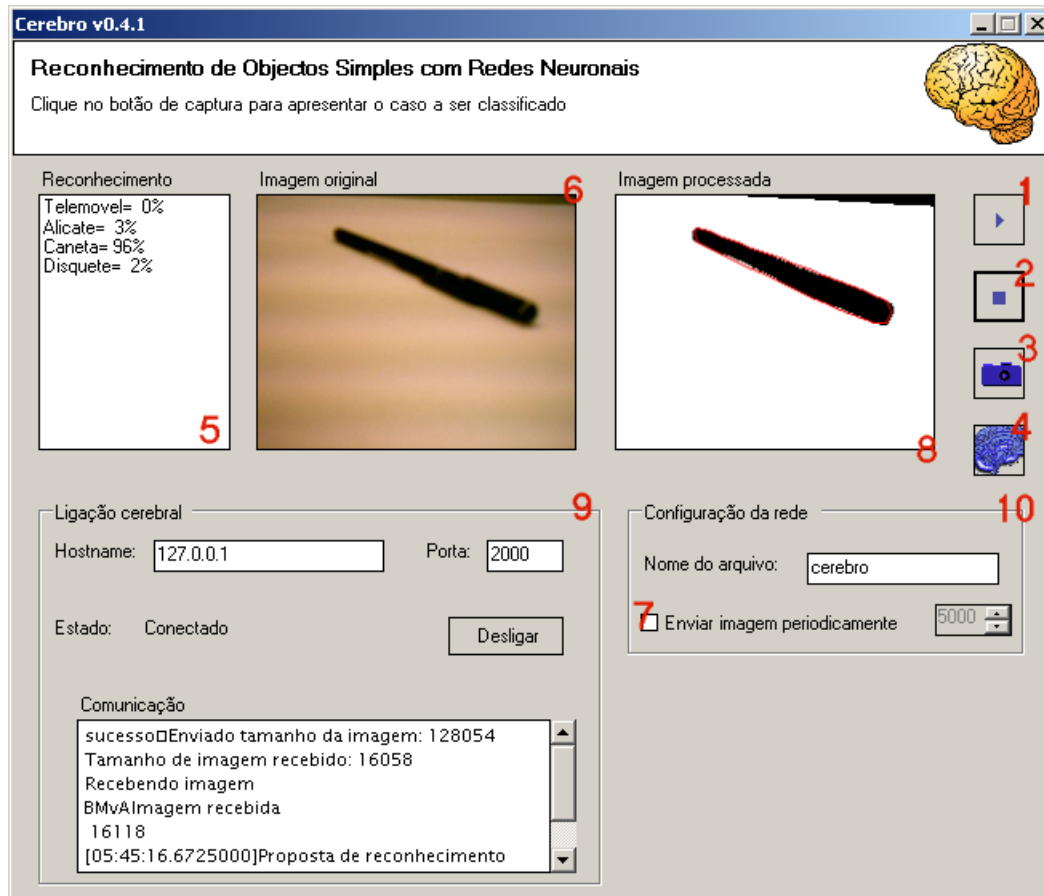
Figura 9: Ecra Principal do Programa

1. Início de captura, este botão inicia a captura da máquina fotográfica digital. Caso não esteja nenhuma máquina ligada, se estiver uma imagem no clipboard, esta será capturada.
2. Parar a captura
3. Enviar a imagem presente no ecrã 6 naquele momento para o servidor, esta imagem será então pré-processada e enviada para o ecrã 8.
4. Processar a última imagem enviada para o servidor (imagem no ecrã 8), a resposta é então enviada para o ecrã 5.
5. Ecrã de reconhecimento de objecto, mostra a percentagem de classificação da imagem relativamente aos objectos que a rede conhece.
6. Imagem capturada pela câmara no momento (ou retirada do clipboard).

7. Automatizar o processo de captura e envio da imagem para pré-processamento
8. Imagem pré-processada, com o contorno detectado ilustrado a vermelho
9. Gestão da ligação ao servidor
10. Nome da rede neuronal a ser utilizada para o reconhecimento